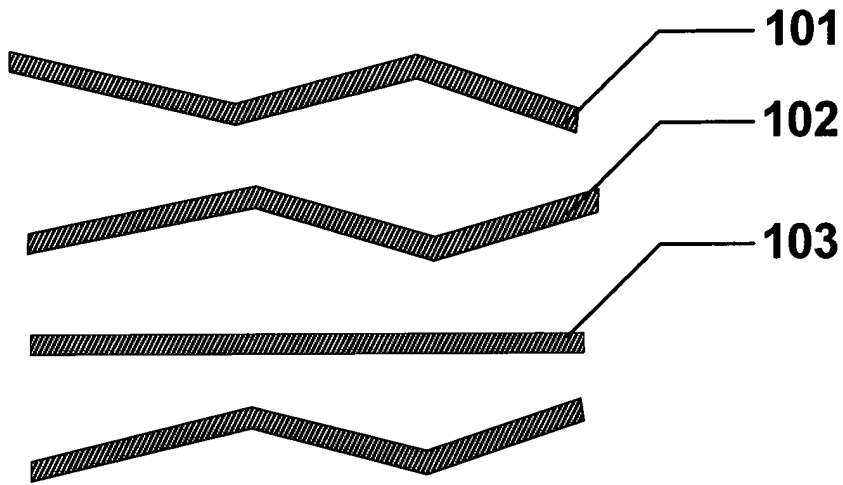
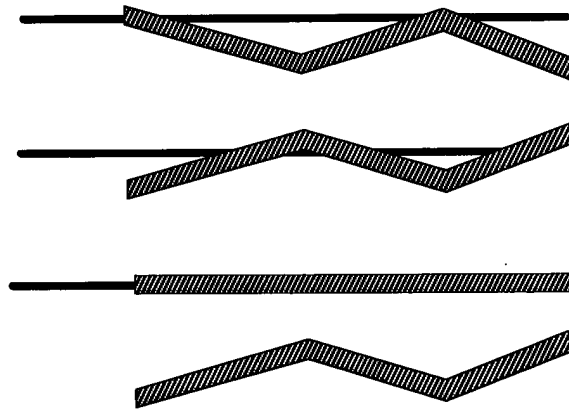


BEST AVAILABLE COPY

1/21

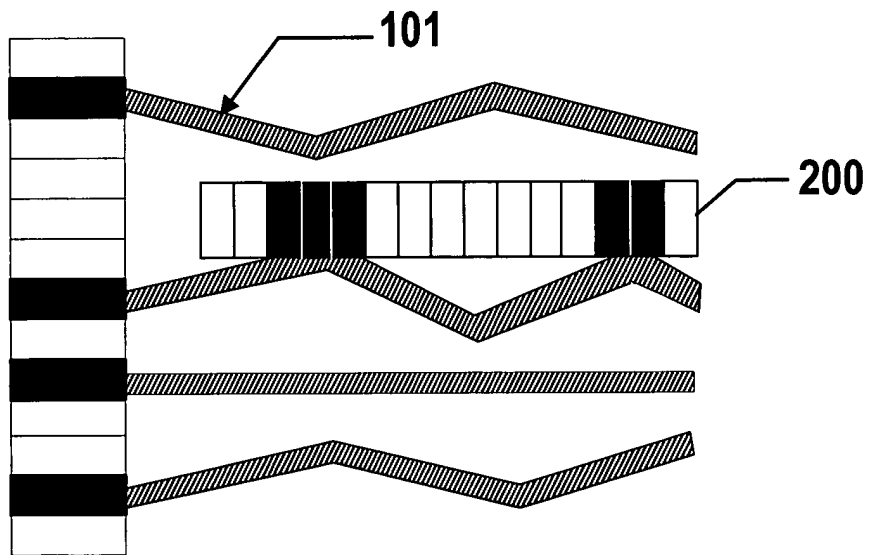
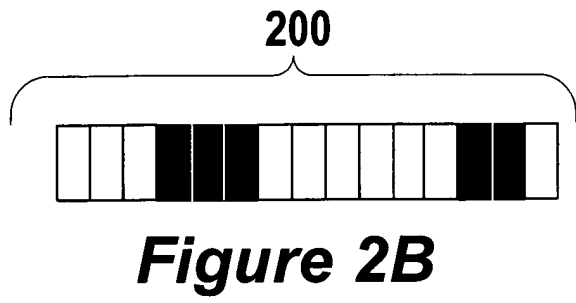
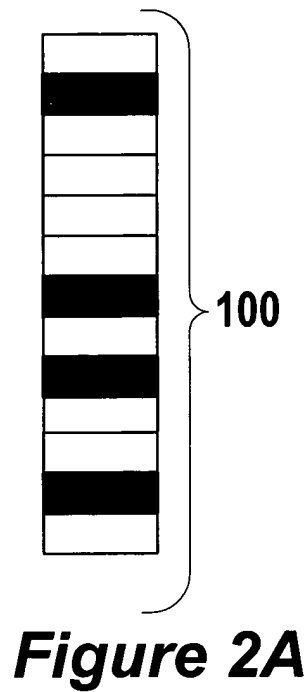


**Figure 1A**

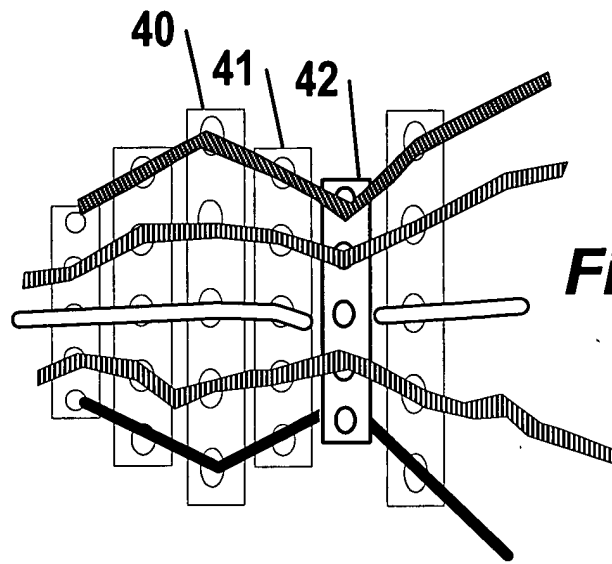


**Figure 1B**

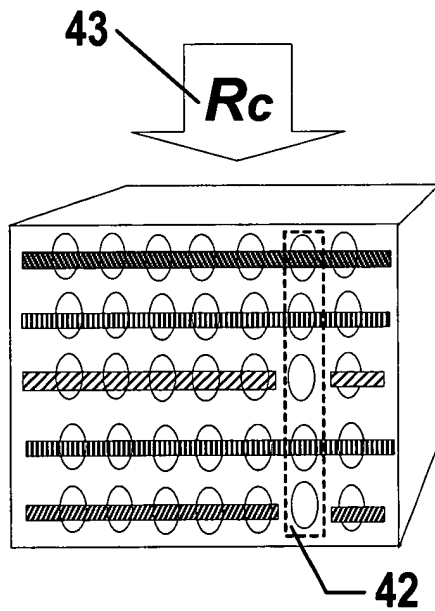
BEST AVAILABLE COPY



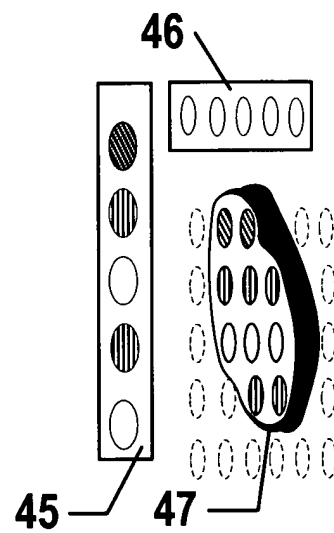
**Figure 3**



**Figure 4A**

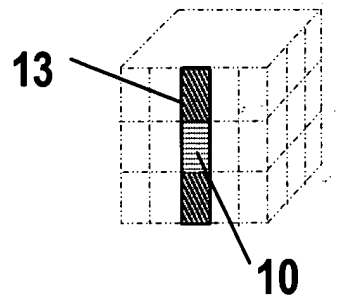


**Figure 4B**

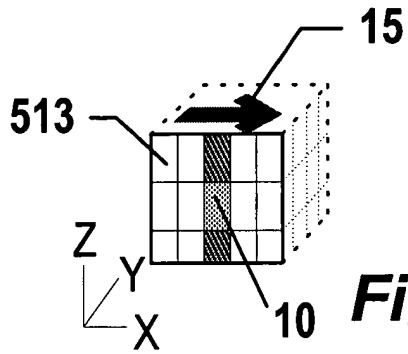


**Figure 4C**

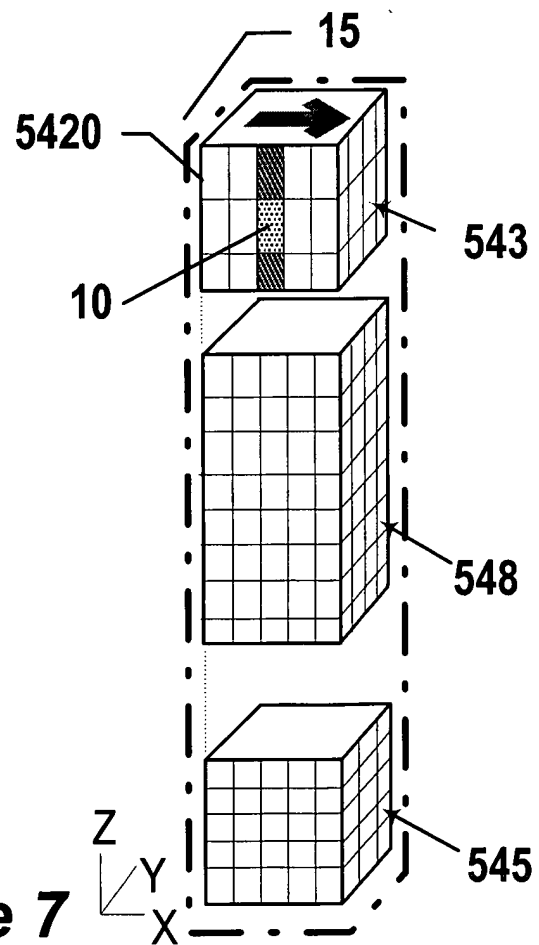
4/21



**Figure 5**

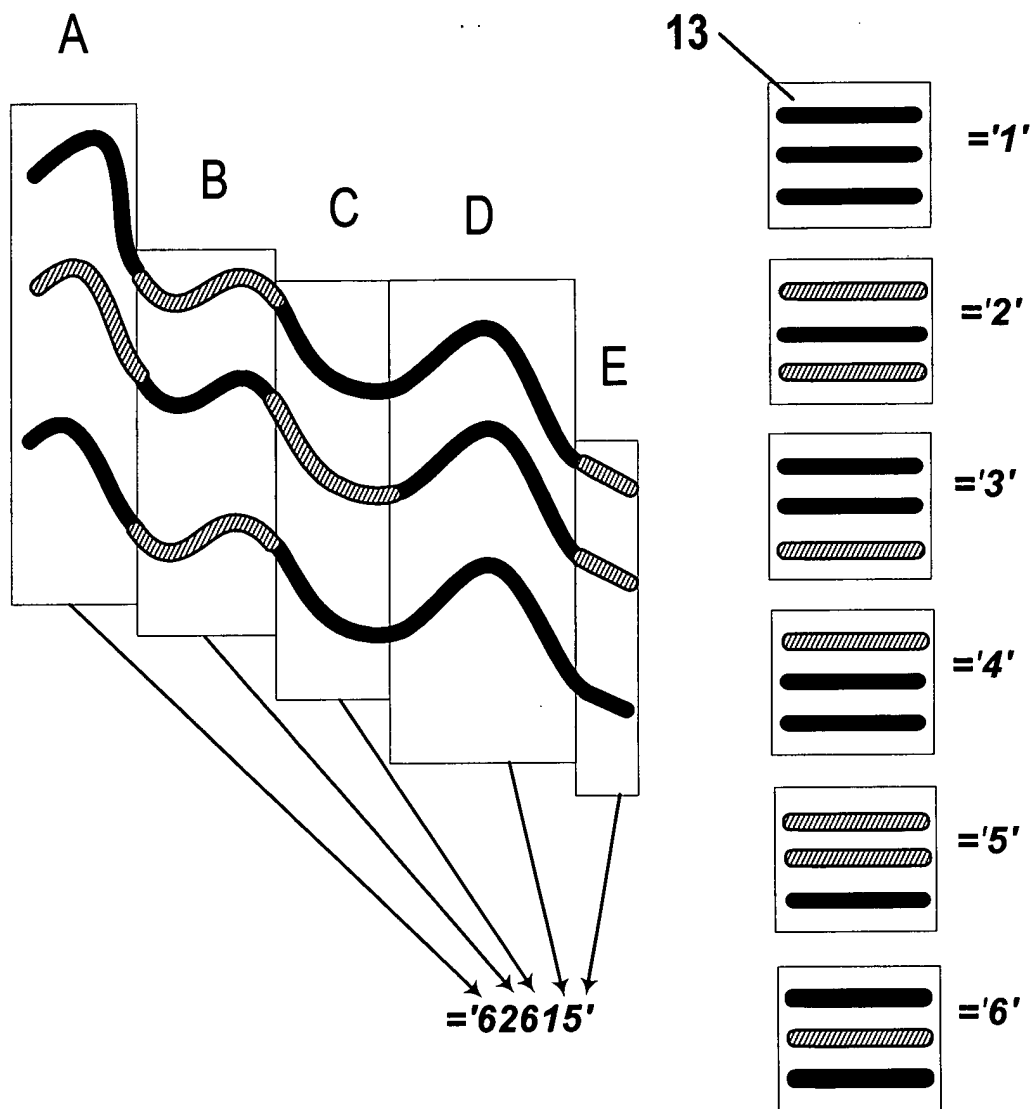


**Figure 6**



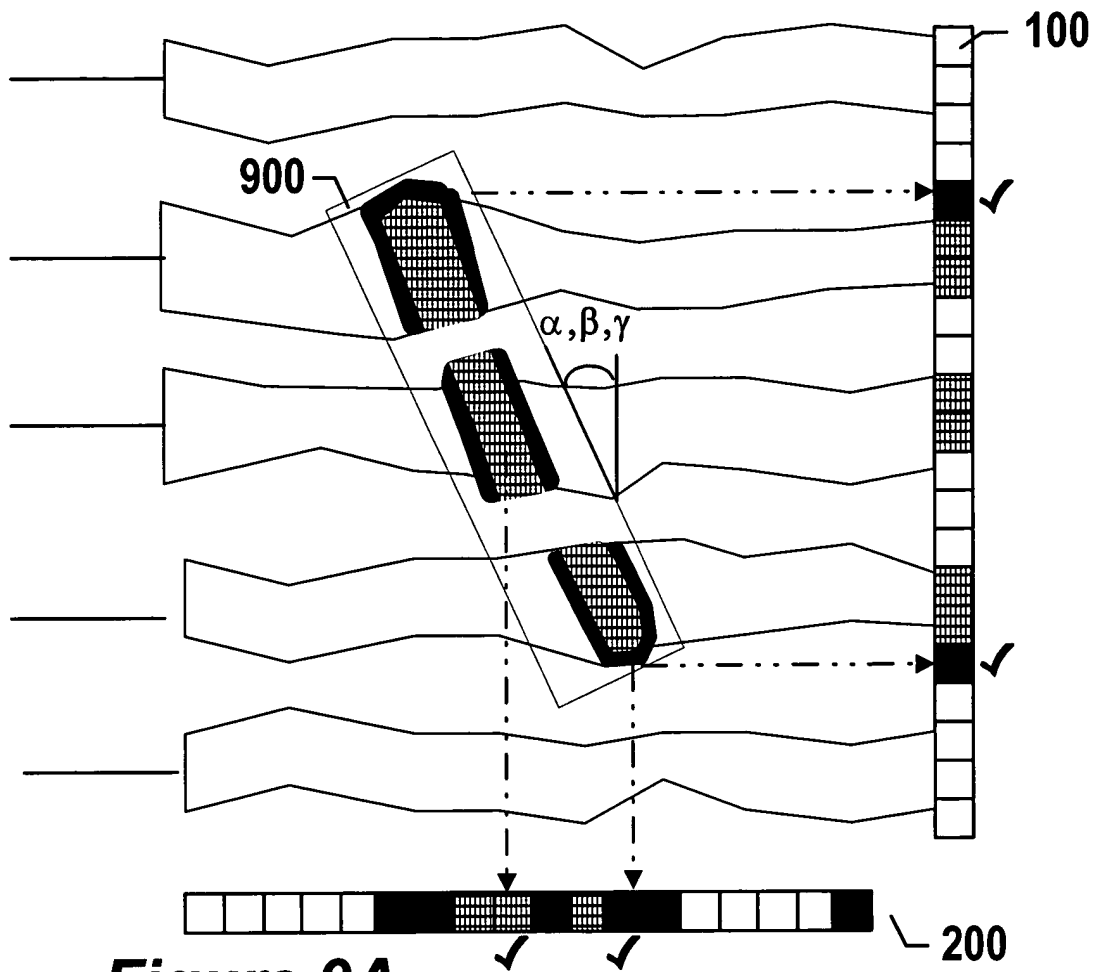
**Figure 7**

5/21

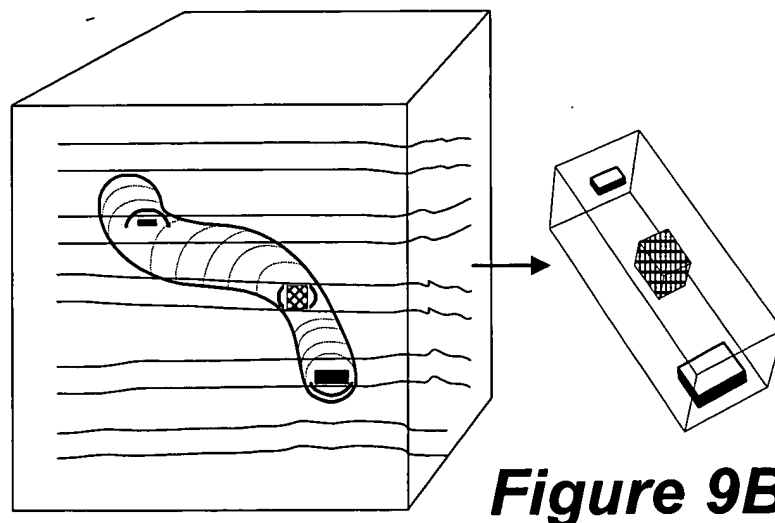


**Figure 8**

6/21

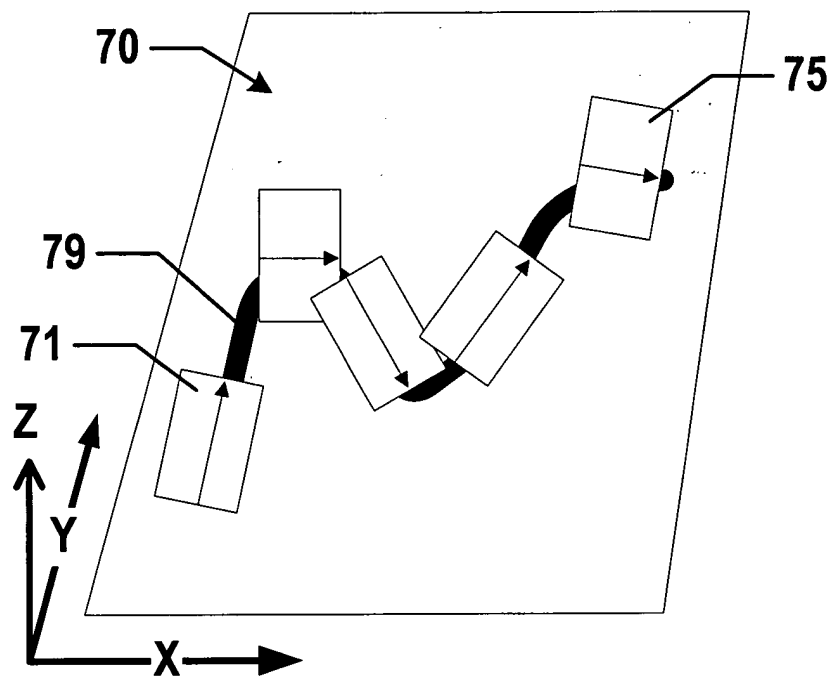


**Figure 9A**

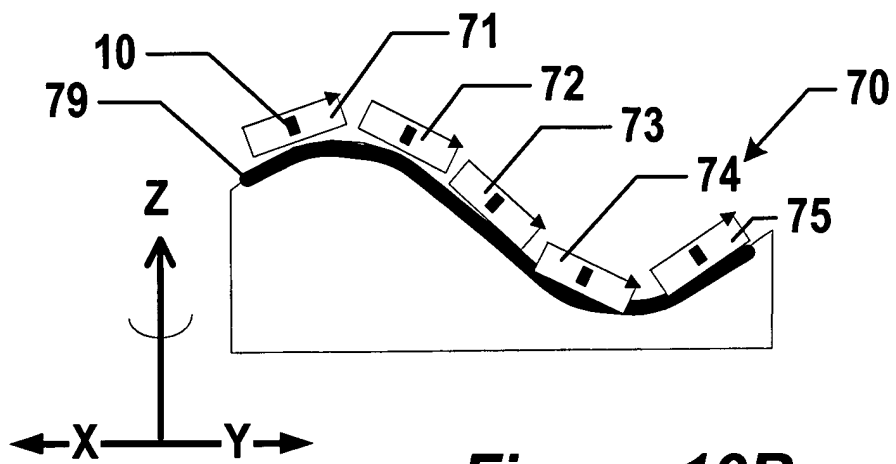


**Figure 9B**

7/21

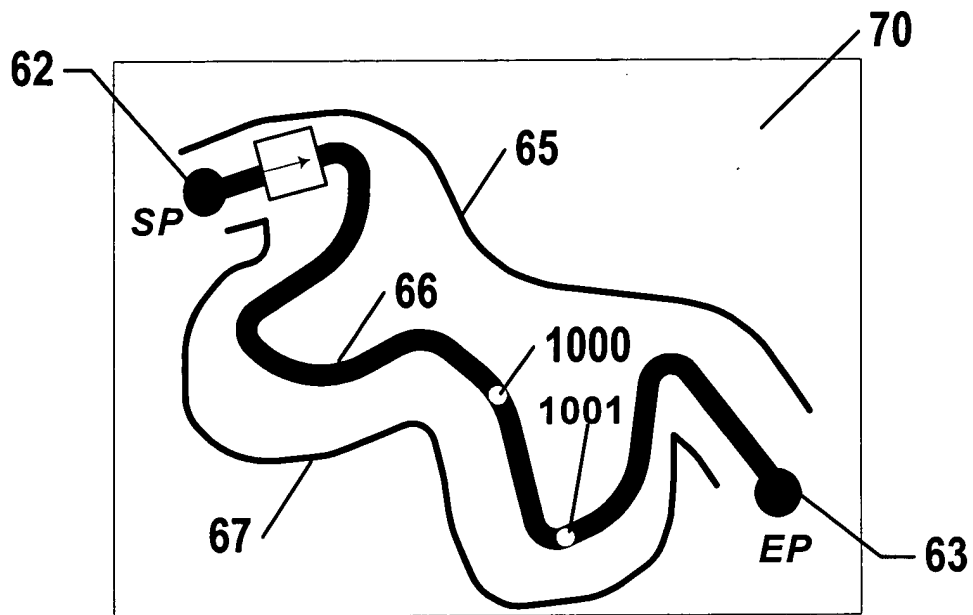


**Figure 10A**

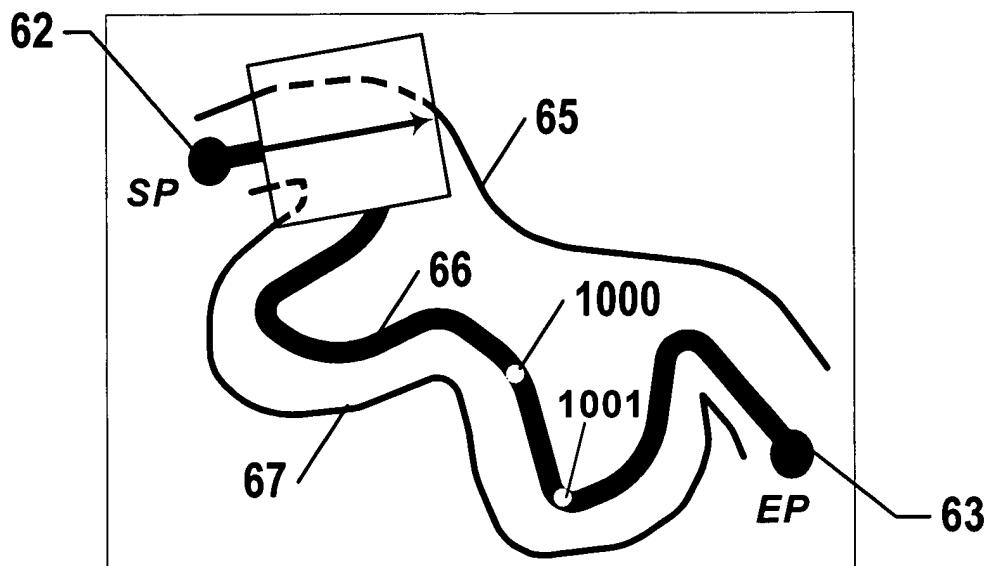


**Figure 10B**

8/21



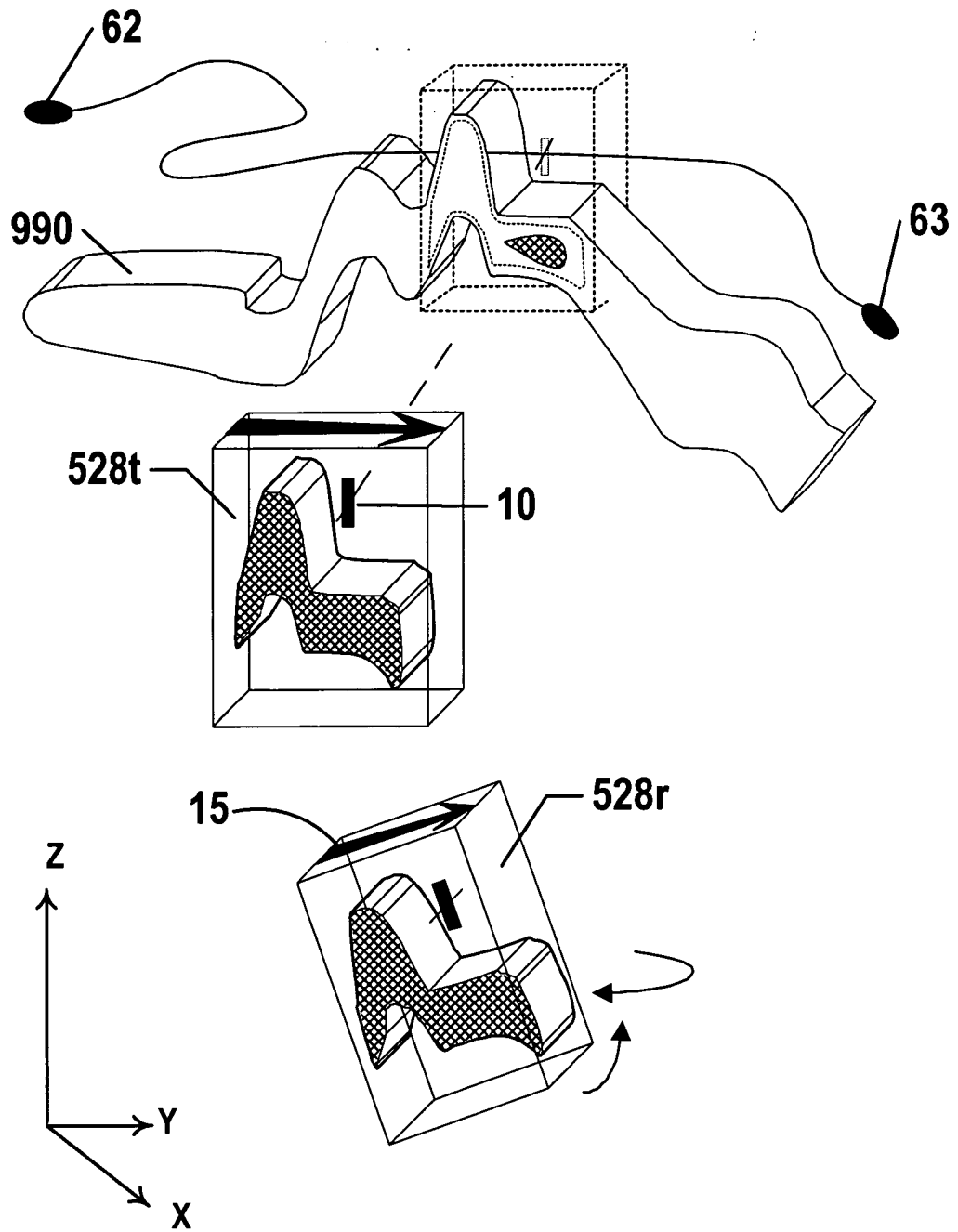
**Figure 11A**



**Figure 11B**

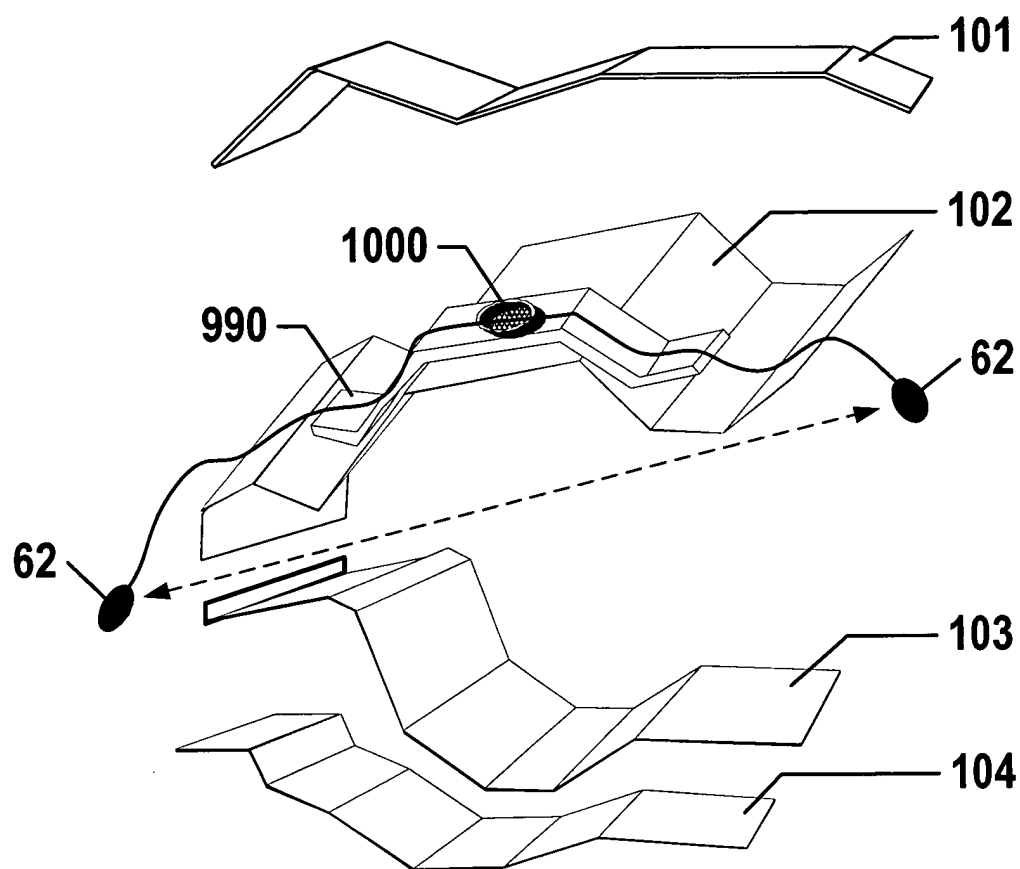


9/21



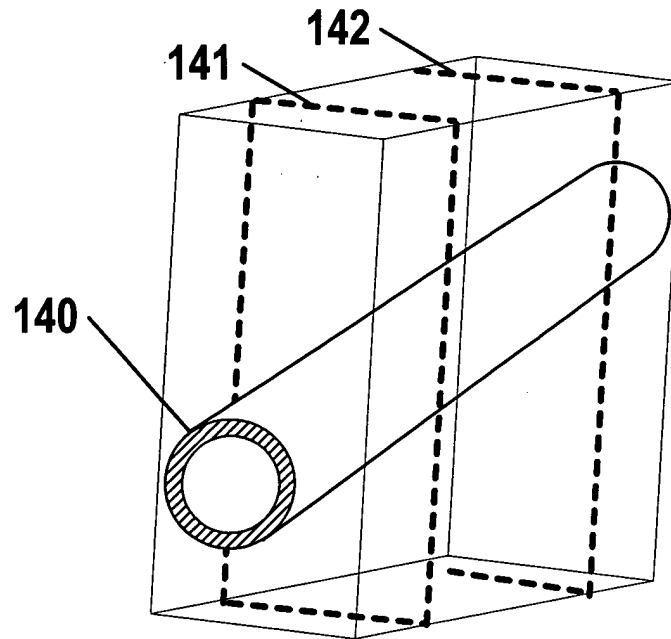
**Figure 12**

10/21

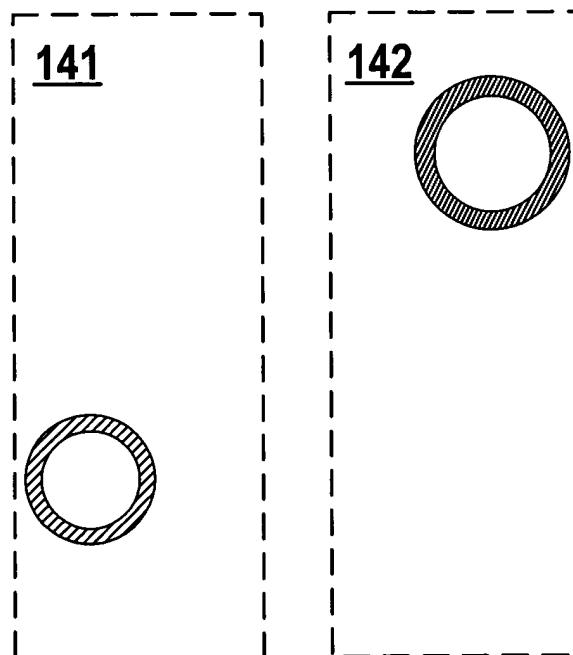


**Figure 13**

11/21

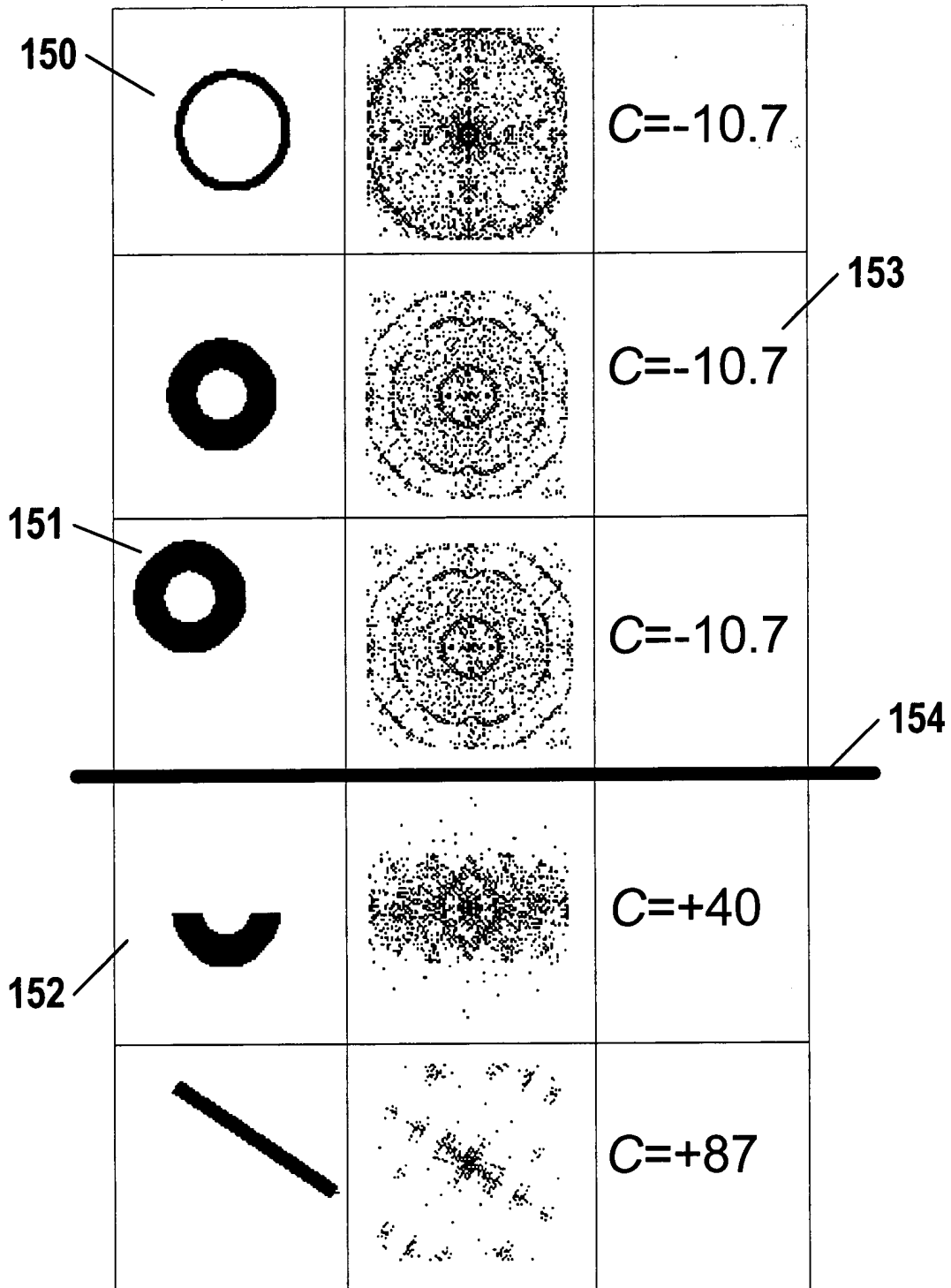


**Figure 14A**



**Figure 14B**

12/21



**Figure 15**

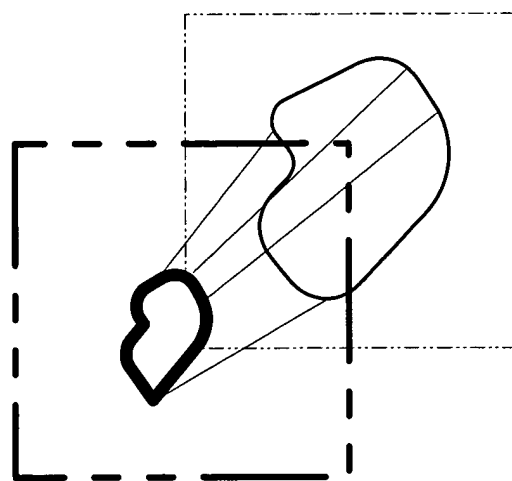
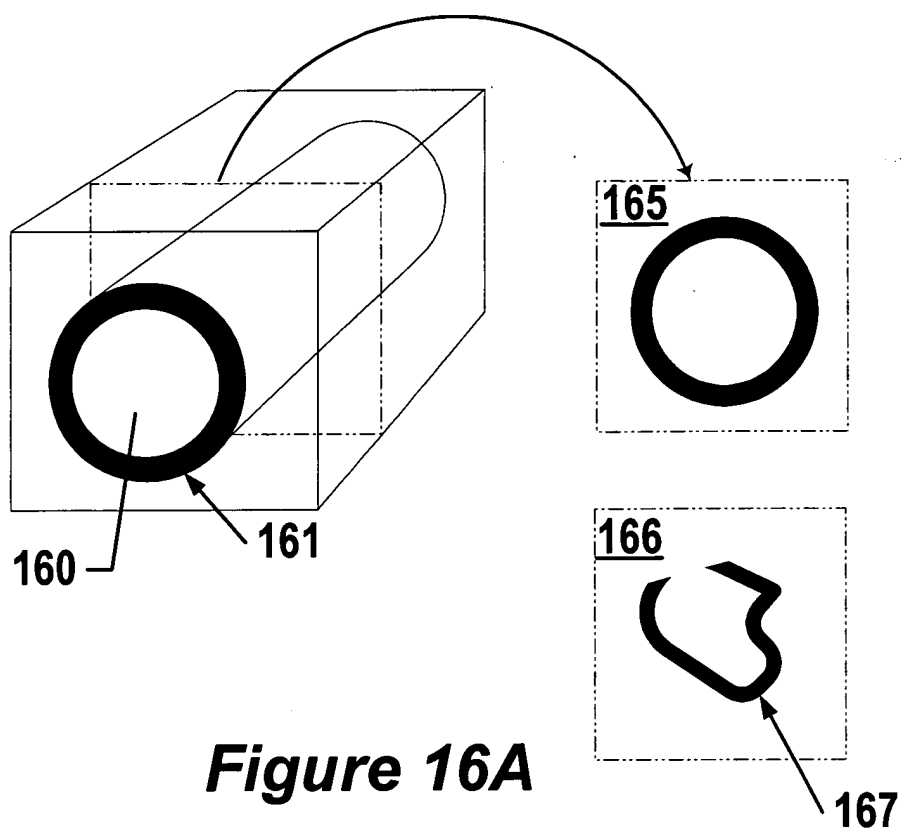
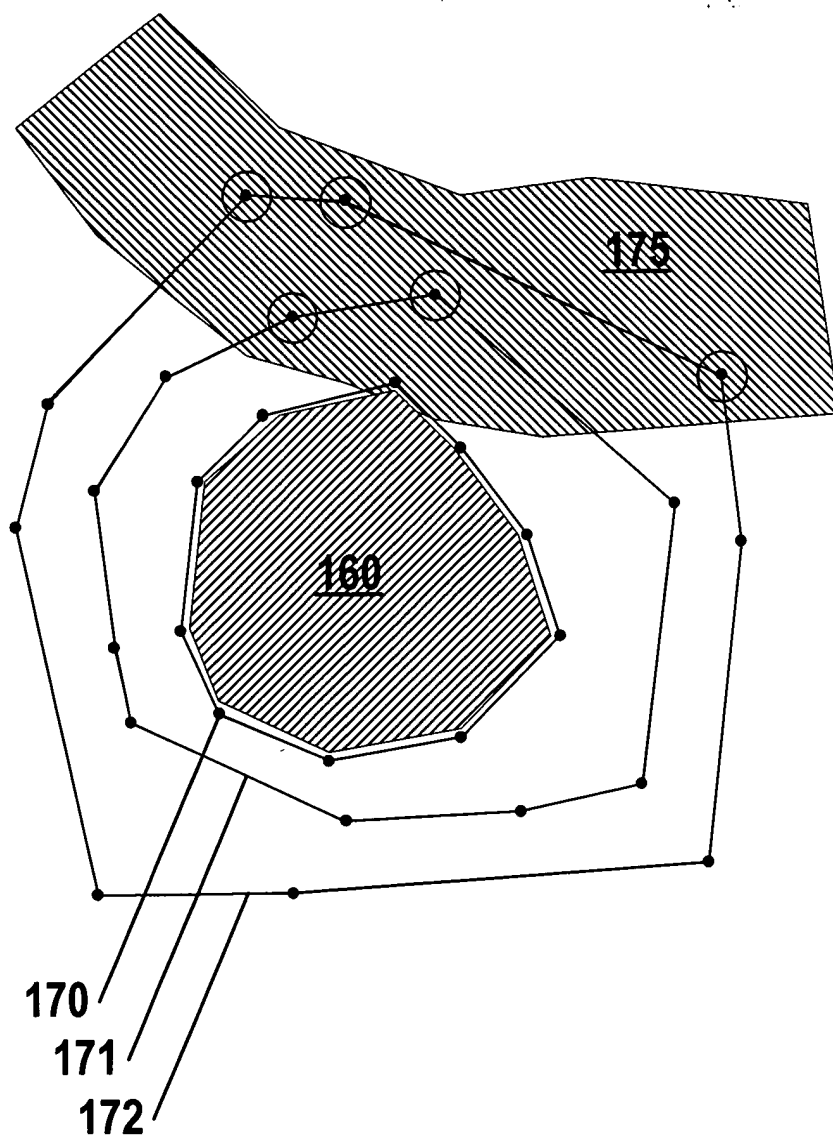
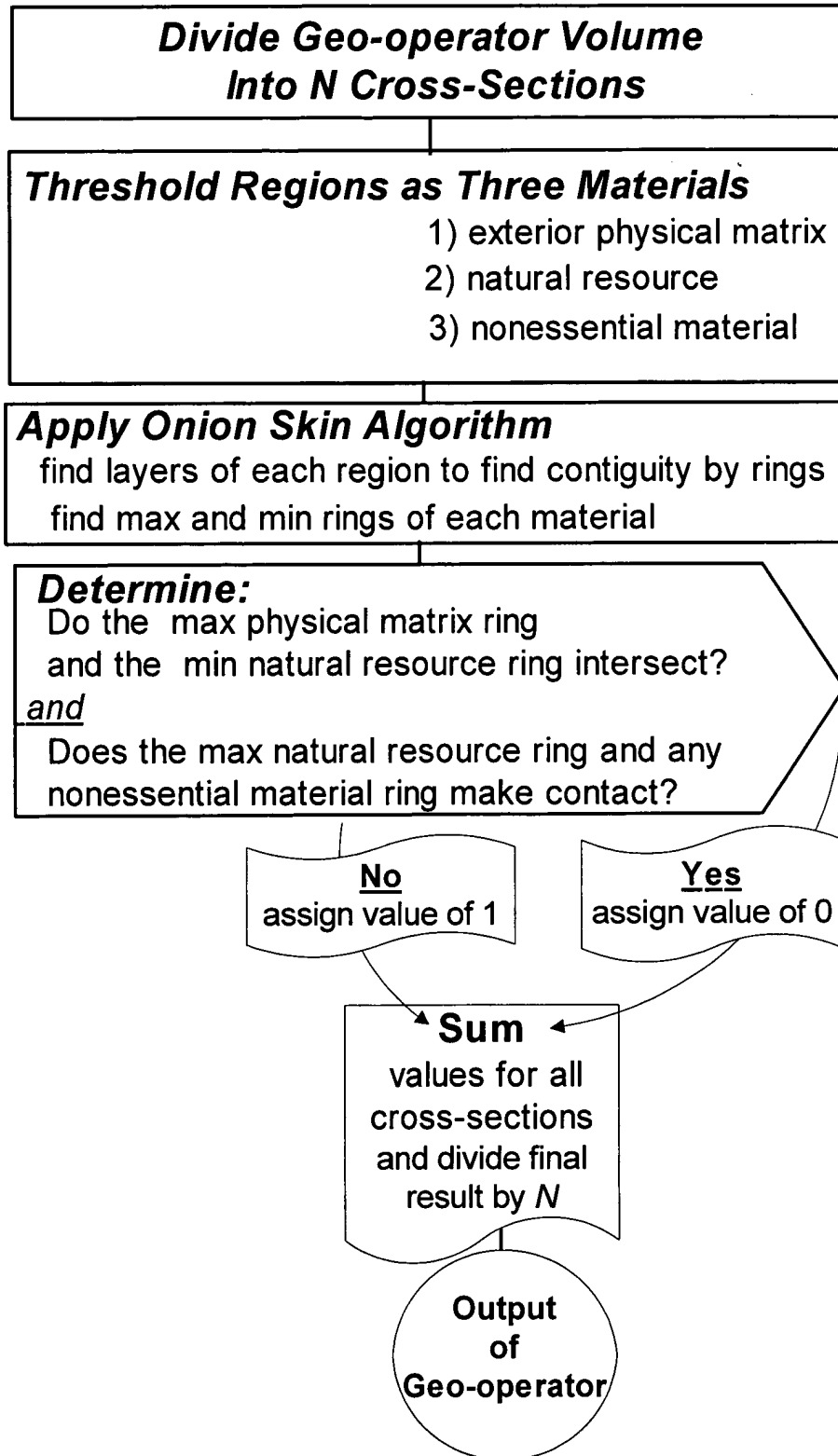


Figure 16B



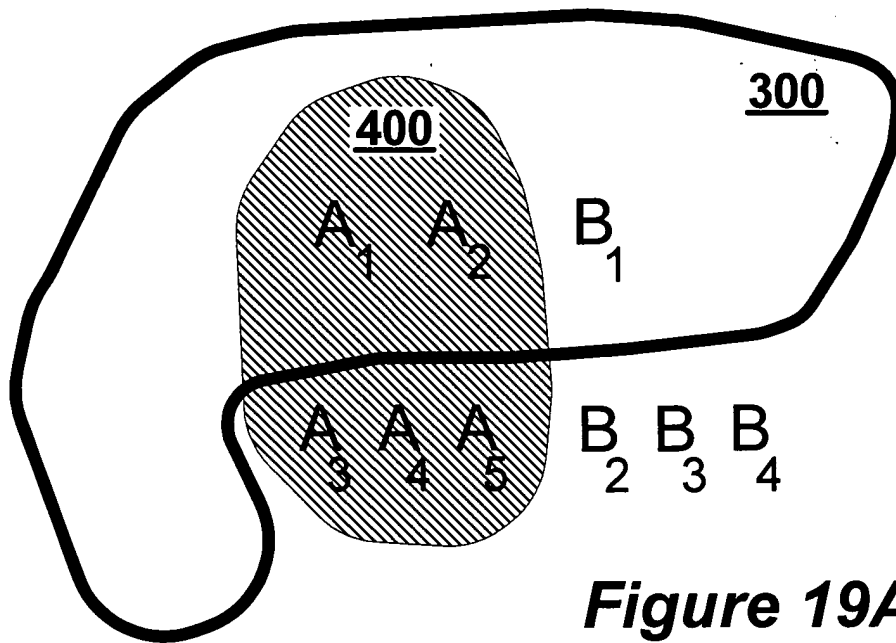
**Figure 17**

15/21

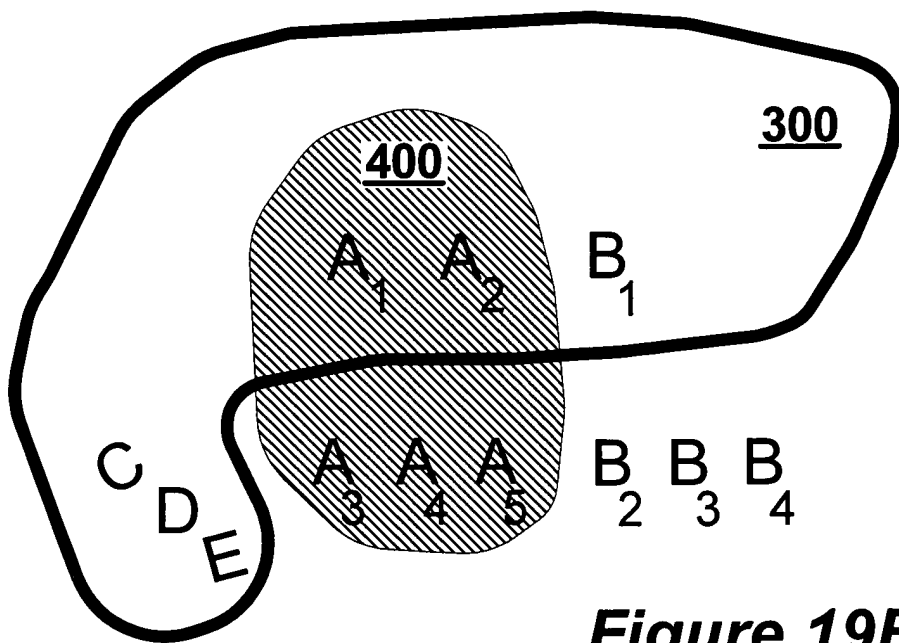


**Figure 18**

16/21



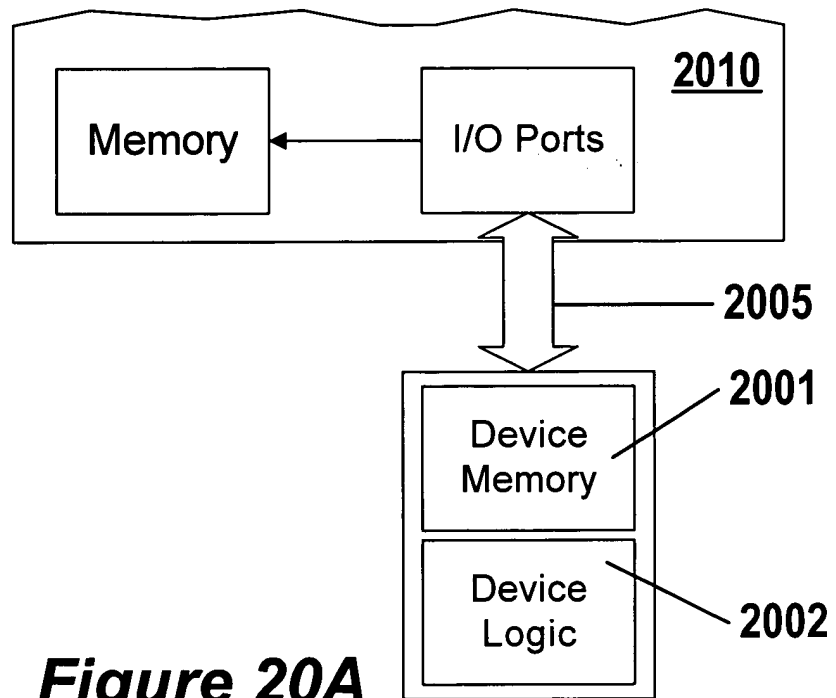
**Figure 19A**



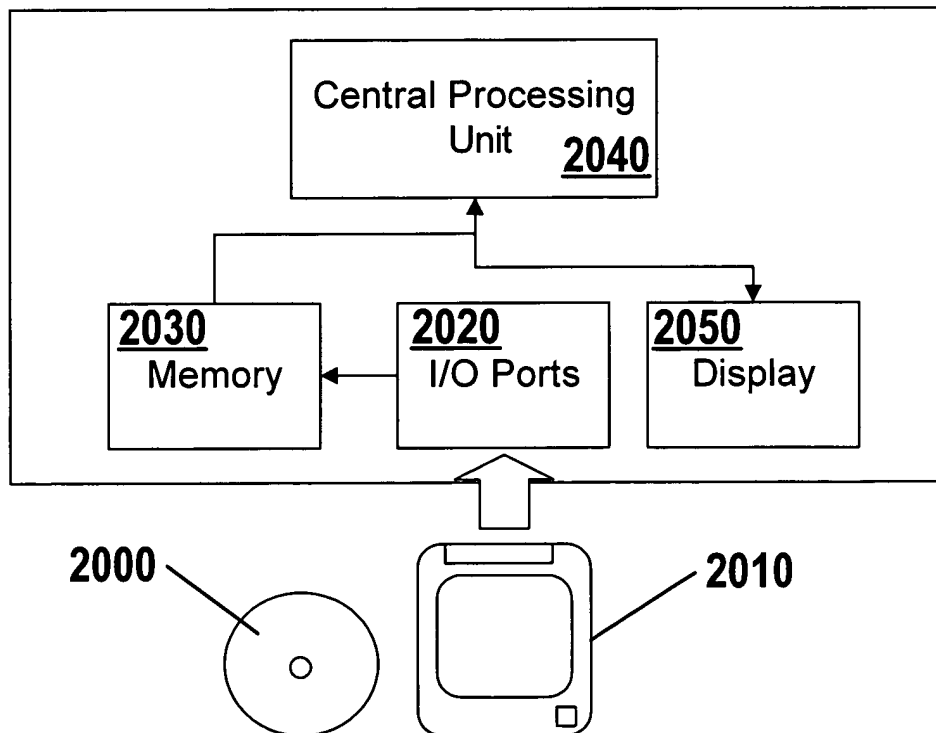
**Figure 19B**



17/21



**Figure 20A**



**Figure 20B**

## PROGRAMNAME FOLLOWER

BEGIN

~COMMENTING Select the path

```

2100 INPUT N, Pathpoints
2110 SET startpoint < Pathpoints(1)
2120 SET endpoint < Pathpoints(N)
2130 SET Refpoint < startpoint
2140 SET voxelDistance < Pathpoints(2)- Pathpoints(1)

```

~COMMENTING Select the geo-operator

```

2200 INPUT xlength, ylength, zlength
2210 INPUT voxrefxyz
2220 INPUT Directionxyz
2230 INPUT Algorithm
2240 INPUT validityInfringementRule
2250 INPUT K, DecisionBoundaryTable
2260 SET opx < xlength
2270 SET opy < ylength
2280 SET opz < zlength

```

2300 LOOPSET M &lt; N ; DOWHILE M &gt; 0

~COMMENTING Move the geo-operator if not at the startpoint

```

2310 IF (Refpoint EQUAL startpoint)
2320 SET Newpoint < Refpoint
OTHERWISE
2330 SET Newpoint < Refpoint + voxelDistance
ENDIF
2340 CALCULATE SmoothRotatedIndices(pathpoints Opx,Opy,Opz, ,Direction,Refpoint,Newpoint)
2350 CALCULATE VerticalFloat(pathpoints,Opx,Opy,Opz, Direction,Refpoint,Newpoint)

```

~COMMENTING Determine if valid computation is possible

```

2400 CALCULATE validityInfringementRule(SmoothRotatedIndices, VerticalFloat, Opxyz, Newpoint)
2410 IF (validityInfringementRule <= 0)
2420 SET voxelValidityAttribute(Pathpoints(M)) < FALSE
OTHERWISE
2430 SET voxelValidityAttribute(Pathpoints(M)) < TRUE
2440 LOOPSET H < size(Opx,Opy,Opz) ; DOWHILE H > 0

```

~COMMENTING Calculate and store the geo-operator

```

2500 INPUT BrickofData(SmoothRotatedIndices)
2510 CALCULATE Algorithm(Pathpoint(M))
2520 STORE CalculationResult(M)
2530 CALCULATE DecisionBoundaryResult(DecisionBoundaryTable(CalculationResult(M)))
2540 SET voxelPathAttribute(Pathpoints(M)) < PathColor
2550 SET voxelCalculationAttribute(Pathpoints(M)) <
DecisionBoundaryResult(M)
2570 STORE voxel(...)Attribute(Pathpoints(M))
2580 DECREMENT H
2590 ENDOFLOOPON H
ENDIF

```

~COMMENTING Increment the position and go to next path point

```

2600 SET Refpoint < Refpoint + voxelDistance
2610 DECREMENT M
2620 ENDOFLOOPON M

```

~COMMENTING Visualize results

```

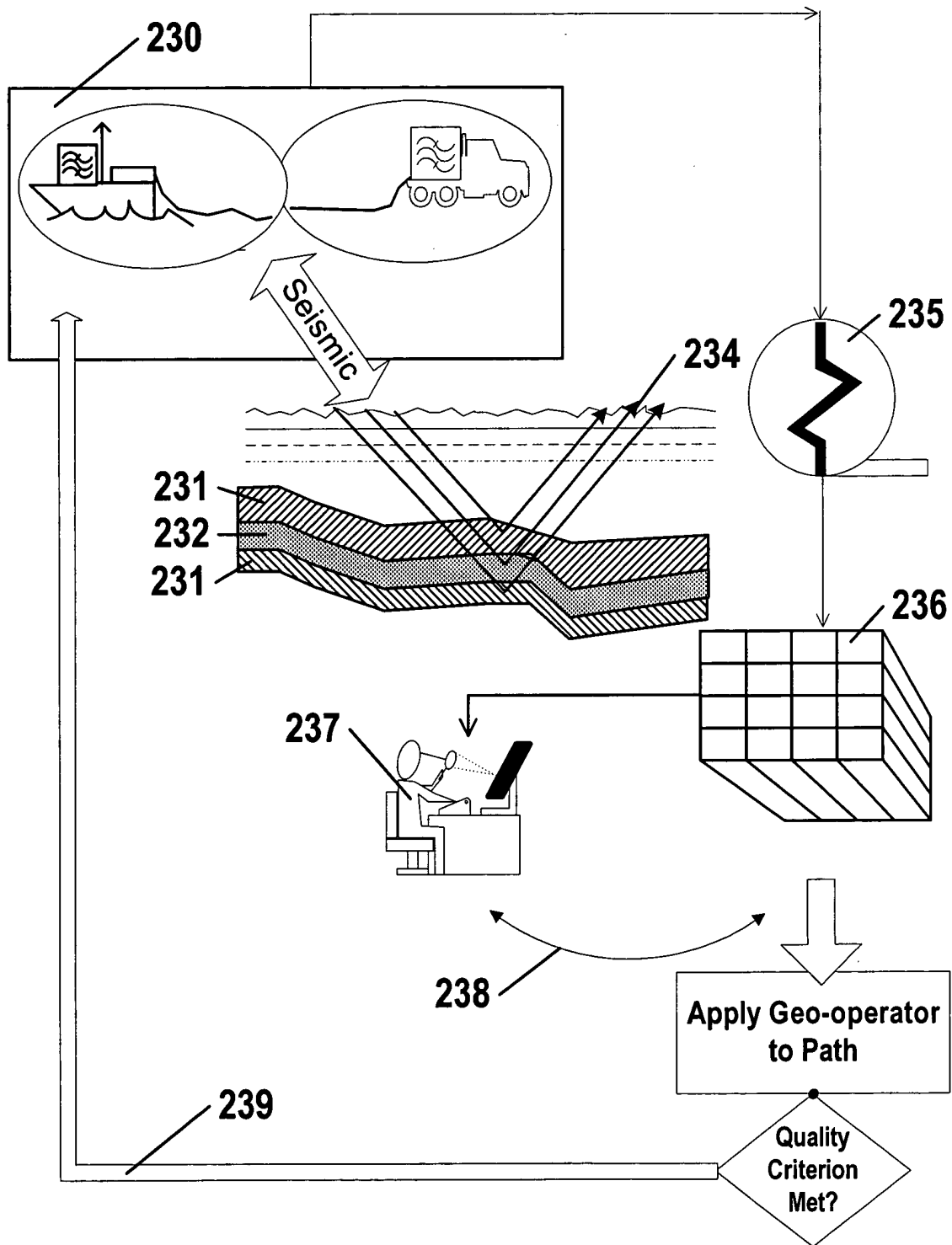
2700 RECALL voxel(...)Attribute(pathpoint(...))
2710 DISPLAY voxel(...)Attribute(pathpoint(...))
ENDPROGRAM.

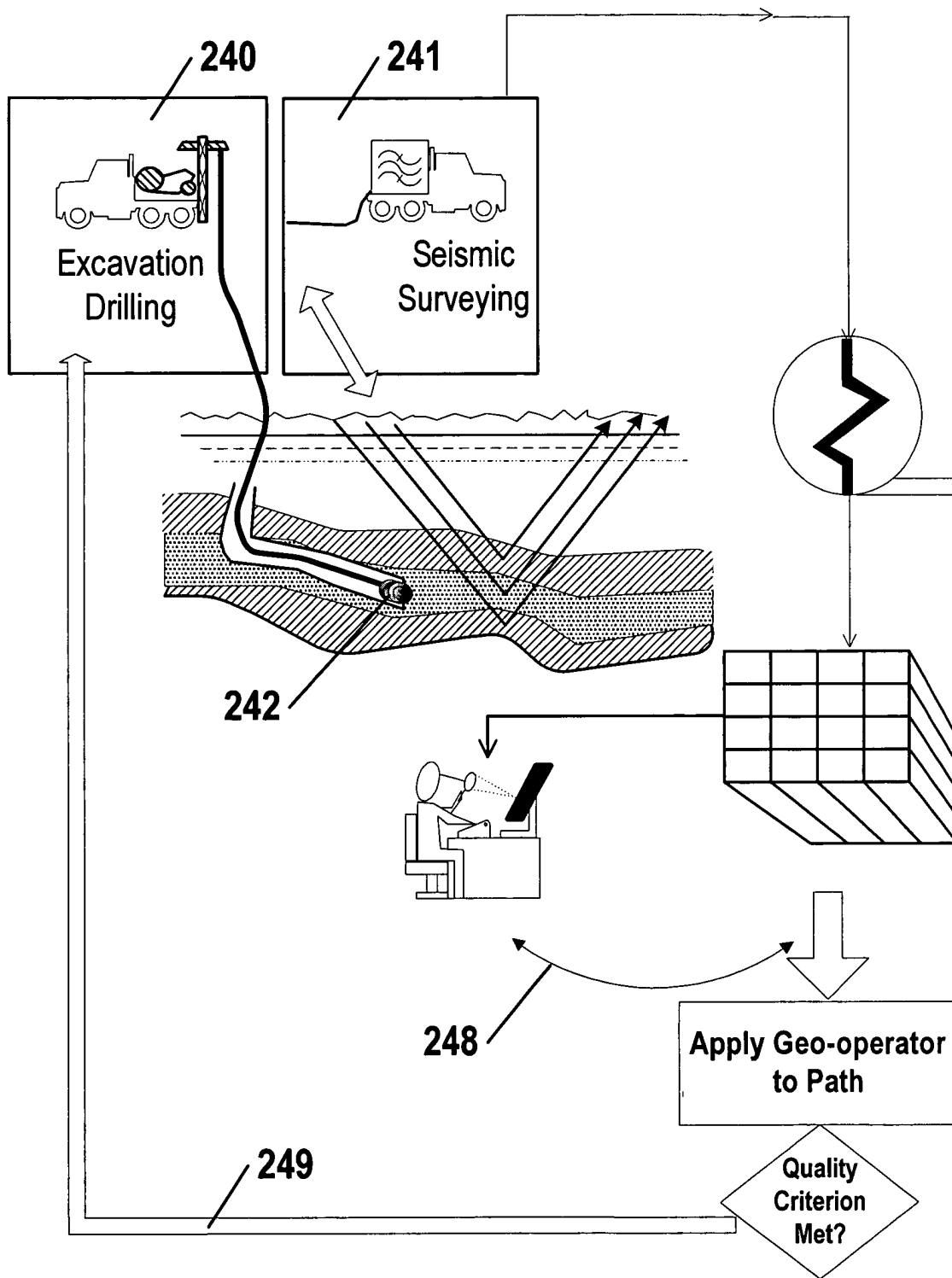
```

Figure 21



**Figure 22**

**Figure 23**



**Figure 24**